GTO : A toolkit to unify pipelines in genomic and proteomic research

Almeida, Joao R.

2020

Almeida , J R , Pinho , A J , Oliveira , J L , Fajarda , O & Pratas , D 2020 , ' GTO : A toolkit to unify pipelines in genomic and proteomic research ' , SoftwareX , vol. 12 , 100535 . https://doi.org/10.1016/j.softx.202

http://hdl.handle.net/10138/324968
https://doi.org/10.1016/j.softx.2020.100535

SoftwareX 12 (2020) 100535

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original software publication

GTO: A toolkit to unify pipelines in genomic and proteomic research

João R. Almeida [a,b,*], Armando J. Pinho [a], José L. Oliveira [a], Olga Fajarda [a], Diogo Pratas [a,c]

[a] Department of Electronics, Telecommunications and Informatics (DETI/IEETA), University of Aveiro, Aveiro, Portugal
[b] Department of Information and Communications Technologies, University of A Coruña, A Coruña, Spain
[c] Department of Virology, University of Helsinki, Helsinki, Finland

ARTICLE INFO

Article history:
Received 26 March 2020
Received in revised form 22 May 2020
Accepted 3 June 2020

Keywords:
Genomic Toolkit
Proteomic Toolkit
Next-generation sequencing

ABSTRACT

Next-generation sequencing triggered the production of a massive volume of publicly available data and the development of new specialised tools. These tools are dispersed over different frameworks, making the management and analyses of the data a challenging task. Additionally, new targeted tools are needed, given the dynamics and specificities of the field. We present GTO, a comprehensive toolkit designed to unify pipelines in genomic and proteomic research, which combines specialised tools for analysis, simulation, compression, development, visualisation, and transformation of the data. This toolkit combines novel tools with a modular architecture, being an excellent platform for experimental scientists, as well as a useful resource for teaching bioinformatics enquiry to students in life sciences. GTO is implemented in C language and is available, under the MIT license, at https://bioinformatics.ua.pt/gto.

Code metadata

Current code version	v1.5
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX_2020_147
Legal Code License	MIT
Code versioning system used	GIT
Software code languages, tools, and services used	C
Compilation requirements, operating environments & dependencies	GCC and Make
If available Link to developer documentation/manual	https://github.com/cobilab/gto/blob/v1.5/manual/manual.pdf
Support email for questions	pratas@ua.pt

Software metadata

Current software version	v1.5
Permanent link to executables of this version	https://anaconda.org/Cobilab/gto
Legal Software License	MIT
Computing platforms/Operating Systems	Linux and Unix-like
Installation requirements & dependencies	GCC and Make
If available, link to user manual - if formally published include a reference to the publication in the reference list	https://github.com/cobilab/gto/blob/v1.5/manual/manual.pdf
Support email for questions	pratas@ua.pt

1. Motivation and significance

Next-generation sequencing (NGS) has become an essential tool in genetic and genomic analysis with a substantial impact in the fields of biomedicine and anthropology. The advantages of NGS over traditional methods include its multiplex capability and analytical resolution, making it a time and cost-efficient approach for fast clinical and forensic screening [1]. The development of efficient bioinformatics tools is essential to assess and analyse the large volumes of sequencing data produced by next-generation

* Corresponding author at: Department of Electronics, Telecommunications and Informatics (DETI/IEETA), University of Aveiro, Aveiro, Portugal.
E-mail address: joao.rafael.almeida@ua.pt (J.R. Almeida).

https://doi.org/10.1016/j.softx.2020.100535

sequencers. However, more important than that are the computational methods that unify the existing tools, given the notable pace at which these tools become available.

Toolkits are sets of tools that combine multiple features in a custom-based manner as some examples show, both in genomics [2] and proteomics [3]. Developing a toolkit requires a specific architecture, namely, taking into account the purpose and technologies, accessibility, compatibility, portability, interoperability, and usability. Moreover, implementation needs to consider efficiency, while maintaining affordable computational resources and the absence of dependencies (standalone use).

We contribute with GTO (Genomic Toolkit), a set of tools to unify pipelines operating both at genomic and proteomic levels, with an open licence and free of any dependency. This toolkit includes information theory-based tools for reference-free and reference-based data compression applied to data analysis. Among many applications, this toolkit supports the creation of workflows for identification of metagenomic composition in FASTQ reads, detection and visualisation of genomic rearrangements, mapping and visualisation of variation, localisation of low complexity regions, or simulation of sequences with specific SNP and structural variant rates. The toolkit was designed for Unix/Linux-based systems, built for ultra-fast computations. It supports pipes for easy integration with the sub-programmes as well as external tools. GTO works as *LEGOs*^{TM}, since it allows the construction of multiple pipelines with many combinations. We support the toolkit with a detailed manual and a website with several examples, including an online manual for fast learning.

Due to the variety and distribution of the given tools and their tight interconnection using the command line with pipes, the toolkit is an excellent platform for scientists as well as for empowering students to progress to the scientific aspects of bioinformatics analysis efficiently. Therefore, without the need to install multiple programmes, dependencies, and read different manuals or licences, it is possible to maintain an easy-to-follow connection with all the phases of each pipeline application.

2. Software description

GTO is a powerful toolkit composed of more than 75 tools with particular focus on genomics and proteomics, following an integrative and flexible design between the tools. GTO includes tools for information display, randomisation, edition, conversion, extraction, search, calculation, compression, simulation and visualisation. The toolkit can be used in common Linux distributions. We have been using GTO in common personal computers (e.g. a laptop with 8 GB RAM, 128 GB of SSD and an intel-i3 CPU from the 5th generation), but these characteristics can vary according to the data size and the execution requirements.

2.1. Software architecture

The tools composing this toolkit aim for key features such as being easy to use, compile and improve and specially designed for work in Unix/Linux command line. These tools can be used in isolation, or combined as one, forming execution workflows. This is technically possible due to the two streams used for the computation, namely the standard input and output. Furthermore, the tools' aggregation is possible with mechanisms for inter-process communication using message passing, provided by the Unix operating system. This creates a chain of processes in which the output of each process is passed directly as input to the subsequent one, as shown in the following example:

```
gto_tool_1 < input | gto_tool_2 | gto_tool_3 > output
```

In addition to the input/output standard streams, some of the tools accept parameterisation through the definition of arguments when executed. There is also a small set of tools in which the input or output does not make sense to be the standard streams and for those the argument definition is considered.

2.2. Software functionalities

The toolkit contains three main groups of tools according to its characteristic: Genomics, Proteomics, and General purpose. The genomics group is subdivided in: FASTQ, FASTA, SEQ (genomic sequences); while the proteomics contains AA (amino acid); the general-purpose tools can be applied to any format sequence.

2.2.1. Genomics

The toolkit allows data conversion between different formats namely FASTQ, FASTA and SEQ. It also provides features for filtering and randomising DNA sequences, as well as for analytic purposes followed by simulations of a generation and alteration nature. The SEQ cluster works directly with the DNA sequences without any standard format. These tools allow data extraction, summary, classification and mathematical operations in the field of information theory. Among many examples, which are better described in the supporting website and manual, the toolkit allows preparations of the reads, namely filtering and trimming, the automatic construction of nucleotide reference databases, and comparative genomics.

2.2.2. Proteomics

The toolkit has a specific cluster of tools designed to group, compress, and analyse amino acid sequences. These tools allow proteomic analysis based on the amino acids properties, such as electric charge (positive and negative), uncharged side chains, hydrophobic side chains and special cases. The toolkit allows translation of codons into amino acids, permits finding approximate amino acid sequences and performing comparative proteomics analysis.

2.2.3. General purpose

This set of tools is complementary to the genomics and proteomics tools, not being designed to work in a specific field, but to assist the pipelines composed of the previously described subsets. These tools provide operations in the symbolical domain, including reversion, segmentation, and permutation; while in the numerical domain they contain tools with low-pass filters (with multiple window types), sum, min and max operations over streams.

2.2.4. External tools

External top-performing tools have been integrated in order to increase the variety of functionalities available. The tools integrated are the following:

- fastp [4]: enables ultra-fast preprocessing and quality control of FASTQ files.
- bfMEM [5]: detects maximal exact matches between a pair of genomes based on bloom filters and rolling hashs.
- copMEM [6]: another tool for computing maximal exact matches in a pair of genomes.
- qvz [7]: implements a lossy compression algorithm for storing quality scores associated with DNA sequencing.
- minicom [8]: a compressor for short reads in FASTQ files that uses large k-minimisers to index the reads.
- SPRING [9]: which is reference-free compression tool for FASTQ files.

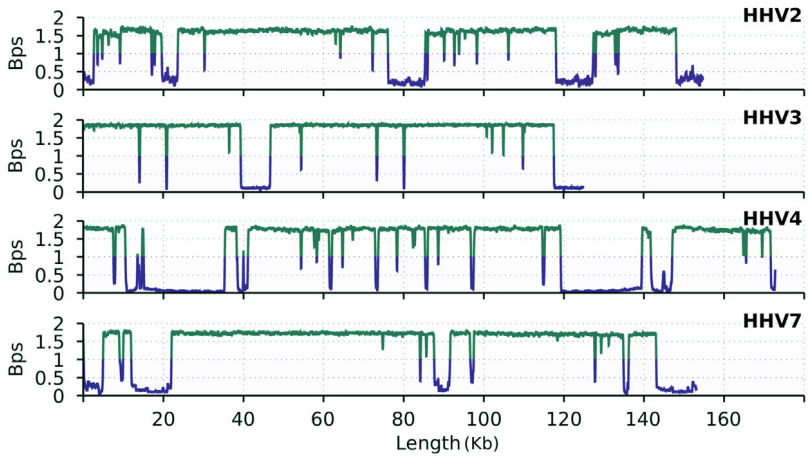

Fig. 1. Bi-directional complexity profiles of four types of human *Herpesvirus* (HHV2, HHV3, HHV4 and HHV7) generated with GTO using the pipeline: `gto_complexity_profile_regions.sh`. Complexity values below one are highlighted with blue colour while the others with green. Bps stands for bits per symbol where lower values represent redundancy. The length is in Kb (Kilobases) and all profiles use the same scale. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

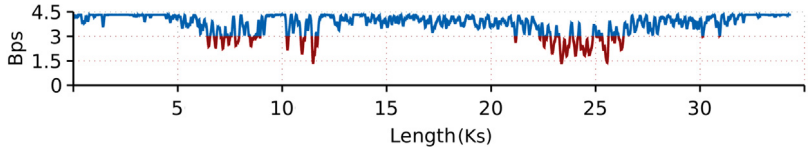

Fig. 2. Bi-directional complexity profiles of human titin protein generated with GTO using the pipeline: `gto_proteins_complexity_profile_regions.sh`. Complexity values below three are highlighted with a red colour while the others with blue. Bps stands for bits per symbol where lower values represent redundancy. The length is in Ks (Kilosymbols). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

3. Illustrative examples

All the tools in the toolkit were tested with synthetic sequences aiming for individual validation. Therefore, the documented examples are easily replicable with the written tests. Besides applying these tools in controlled environments, the toolkit was also used in several research workflows both as a primary and auxiliary tool. Several complete workflows are available in the repository, under the pipelines folder while an extensive description of the tool can be found in the manual. Next, we include some pipeline examples.

3.1. Bi-directional complexity profiles

A workflow example is the computation of bi-directional complexity profiles in any genomic or proteomic sequence [10]. These profiles can localise specific features in the sequences, namely low and high complexity sequences, inverted repeats regions, tandem duplications, among others. The construction of these profiles follows a pipeline formed of many transformations (e.g. reversing, segmenting, inverting) as well as the use of specific low-pass filters after data compression applications [11]. Fig. 1 depicts the complexity profiles of four human *Herpesvirus* whole genomes using the same scale, where redundant regions are highlighted in blue (below a Bps of one).

GTO uses GeCo2 [12] and AC [13] compressors to estimate the local complexity of DNA and amino acid sequences, respectively. However, GTO is not limited to using these data compressors. For example, new models can be tested under this framework, namely with extended alphabets [14]. In general, any data compressor able to output local estimations can be used in the pipeline as an alternative [15].

Analogous to the complexity profiles for DNA sequences, an example using amino acid sequences is given in Fig. 2. This example depicts a bi-directional complexity profile for the largest human protein sequence, titin. Several regions with low complexity are usually associated with specific characteristics, namely loops [16].

3.2. Rearrangements map generation

Another example workflow is in the domain of comparative genomics, namely to map and visualise rearrangements. This workflow is completely automatic from the input of the sequences to the generation of an SVG image, with the associated and transformed regions corresponding to the rearrangements. The pipeline applies smash technology [17,18] for mapping the rearrangements using an alignment-free methodology [19]. To prove the efficiency of the mapping pipeline, we use another pipeline to generate two identical FASTA files with simulated rearrangements between them (`gto_simulate_rearrangements.sh`). After, loading the two FASTA files into the mapping pipeline (`gto_map_rearrangements.sh`), the output is two files, one with the mapping positions and the other is an SVG image depicting the mapped positions as can be seen in Fig. 3. All the rearrangements have been efficiently mapped with GTO according to the ground truth ($<$ 1 s of computational time).

Analogous to the rearrangements map pipeline, for mapping at proteomic level, we consider the NAV2 HUMAN Neuron navigator 2 and the neuron navigator 2 isoform X15 of *Macaca mulatta* proteins. Although there are many examples under the proteome evolution [20], these are protein sequences considering identical scale [21]. Additionally, we shuffled the *Macaca mulatta* proteins using a block size of 300 amino acids. Fig. 4 depicts the proteins map after running the pipeline (`gto_map_rearrangements_proteins.sh`). Despite a low level of dissimilarity of the sequences with an additional pseudo-random permutation of blocks of 300 symbols, all the regions have been efficiently mapped with GTO ($<$ 1 s of computational time).

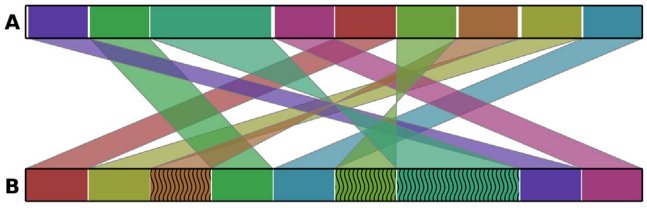

Fig. 3. Rearrangements map generated with GTO using the pipeline: `gto_map_rearrangements.sh`. The length of both sequences (A and B) is 5 MB. Wave pattern stands for inverted repeated regions.

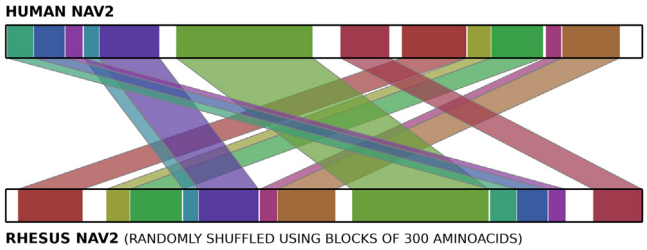

Fig. 4. Rearrangements map generated with GTO using the pipeline: `gto_map_rearrangements_proteins.sh`.

3.3. Viral metagenomic identification

A final workflow example is the full automatic metagenomic identification of viral (or any other) content in FASTQ reads. This includes the filtering and trimming of the reads, mapping, and sensitive identification of the most representative genomes, under a ranking of abundance. In this particular example, we generate a semi-synthetic viral dataset containing several real viruses with applied degrees of substitutions and block permutations shuffled with synthetic noisy DNA. This dataset is generated using the `gto_create_viral_dataset.sh` pipeline.

The intention is to perform a metagenomic analysis on this dataset without informing the programme what organisms are contained in the sample since the programme needs to infer the results. Then, we compare the results with the ground truth. If the results are similar to the ground truth, then the methodology is validated. For the purpose, GTO uses falcon-meta technology [22,23] that relies on assembly-free and alignment-free comparison of each reference according to the whole reads. The dataset contains synthetic reads (uniform distribution) merged with the following viruses with the respective modifications:

- B19V: two *Parvovirus*, one with 1% of editions and the other with permuted blocks of 500 bases (GID: AY386330.1);
- HHV2: one human *Herpesvirus* 2 with permuted blocks of size 100 bases (GID: JN561323.2);
- HHV3: one human *Herpesvirus* 3 (GID: X04370.1);
- HHV4: two human *Herpesvirus* 4, one with permuted blocks of 300 bases (GID: DQ279927.1);
- TTV: one human *Torque teno* virus with 5% of editions (GID: AB041963.1);
- HPV: one human *Papillomavirus* with 5% of editions and permuted blocks of 300 bases (GID: MG921180.1).

After merging all FASTA sequences, ART [24] was used to generate the paired end FASTQ reads. Meanwhile, another workflow example was used to create the viral database (`gto_build_dbs.sh`). Then, the pipeline (`gto_metagenomics.sh`) ran, obtaining the top output presented in Table 1.

We can conclude that despite the noise, editions, and permutations applied to real data, all the viruses have been efficiently identified with GTO, including the exact genotype ($<$ 1.5 min of computational time).

Table 1
The eight most representative reference sequences according to the RS (Relative Similarity). ID stands for the order of the top output, length for the size of the reference genome, and GID for the sequence global identifier.

ID	Length	RS (%)	Reference GID	Virus name
1	124 884	97.767	X04370.1	HHV3
2	5596	96.603	AY386330.1	B19V
3	172 764	94.143	DQ279927.1	HHV4
4	154 675	81.400	JN561323.2	HHV2
5	154 746	80.153	Z86099.2	HHV2
6	2785	78.300	AB041963.1	TTV
7	7372	71.445	MG921180.1	HPV
8	549	47.591	AY034056.1	PHV3-BALF1-gene

4. Impact

Many software application exist to analyse and manipulate sequencing data, namely `fqtools` [25], `GALAXY` [26], `FASTX-Toolkit` [27], `SeqKit` [28], `GATK` [29], among others. The `fqtools` is a suite of tools to view, manipulate and summarise FASTQ data [25]. This software was designed to work specifically with FASTQ files and can be easily integrated into our toolkit. However, the features existent in this software are similar to some of the ones that GTO has in the `gto_fastq_*` section. Both were written in C which in terms of performance could be similar.

`GALAXY`, is an open and web-based scientific platform for analysing genomic data [30]. This platform integrates several specialised sets of tools, e.g. for manipulating FASTQ files [31]. In this web application, the `FASTX-Toolkit` was integrated, which is a collection of command-line tools to process FASTA and FASTQ files [27]. The available features in the `FASTX-Toolkit` are also similar to some of the GTO tools designed to preprocess the FASTA/FASTQ files, which are available in the `gto_fastq_*` and `gto_fasta_*` sections. As our goal always was to have an easy to use toolkit written in low-level programming languages and not a web interface, we cannot compare it with `GALAXY`. However, regarding the `FASTX-Toolkit` which was also written in C, it is possible to compare and combine it with some of the GTO's features.

The `SeqKit` is another toolkit used to process FASTA and FASTQ files and it is available for all major operating systems [28]. Comparing the performance and limitations of this toolkit with the `fqtools` and `FASTX-Toolkit` is easier than comparing them with GTO, mainly because these three toolkits were designed specifically to manipulate FASTA/FASTQ files. On the other hand, these functionalities are only a fraction of the features that we provide in GTO. The idea never was to create more tools to compete with the ones existing, but instead, aggregate them in order to obtain a more complete toolkit for genomics analysis.

This idea of simplifying the development and aggregation of analysis tools for genomic manipulation and analysis is not new. Initially designed as a structured programming framework, the Genome Analysis Toolkit (`GATK`) is a set of bioinformatics tools for analysing high-throughput sequencing focused on variant discovering and genotyping [2]. The high performance of this toolkit is due to the required infrastructures that a personal computer cannot offer. This is an excellent toolkit that integrates Apache Spark for optimisation, but it is only possible to take advantage of this potential in cloud computing.

The efficient performance from some of the presented tools as well as GTO's tools is due to the use of low-level programming languages (e.g. C language). However, one limitation of this strategy, in which the performance is prioritised, is the lack of

a graphical user interface. Moreover, to take full advantage from those tools, the end-users need to have basic shell script knowledge. Nevertheless, GTO combines specialised tools for analysis, simulation, compression, development, visualisation, and transformation of data. Therefore, we would like to highlight some important details that characterise this toolkit:

- The toolkit aggregates different tools in order to build research pipelines to deal with very large data sets without losing performance due to its modular architecture. Adoption of standard streams to interconnect the tools improved data processing. Throughout this procedure, the disk read/write operations between tools have been removed by sending the output directly to the input of the next tool.
- The toolkit can integrate external tools, besides the ones already available. As such, some specific tools that have already been evaluated and used outside this context were aggregated:
  - For compression purposes, the toolkit integrates GeCo2 [12], which along with HiRGC [32], iDoComp [33] and GDC2 [34] are considered to have some of the best performance for reference-free DNA compression [35]. Regarding the amino acid sequences, the toolkit uses the AC tool for lossless sequence compression. The performance of AC was compared in [36] to several general-purpose lossless compressors and several protein compressors using different proteomes and AC provides on average the best bit-rates.
  - Concernings simulation, GTO integrates XS [37] which is a FASTQ read simulation tool. Escalona et al. [38] reviewed 23 NGS simulation tools and XS stands out in relation to the others because it is the only one that does not need a reference sequence.
  - Additionally, we added a section in the toolkit specially designed for tools from other authors. This way, we simplify their integration and installation using GTO. Those were described in Section 2.2.4.
- Finally, as briefly presented in Section 3, the toolkit can answer new genomics questions without the need to create new software.

5. Conclusions

We contribute with GTO, a toolkit to unify research pipelines, composed of distinct tools aiming at efficient combinations of them towards specific workflows. GTO's efficient performance is due to the use of low-level programming languages, which increases the processing speed and decreases the RAM of addressing genomics and proteomics data. The flexibility of this toolkit allows the end-user to quickly create new processing pipelines in the genomic and proteomic field as it was described in the examples provided in this manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work has received support from the NETDIAMOND, Portugal project (POCI-01-0145-FEDER-016385), co-funded by Centro 2020 program, Portugal 2020, European Union, and from the FCT - Foundation for Science and Technology, Portugal, in the context of the project UIDB/00127/2020. João Almeida is supported by FCT - Foundation for Science and Technology, Portugal (national funds), grant SFRH/BD/147837/2019.

References

[1] Mardis ER. DNA sequencing technologies: 2006–2016. Nat Protoc 2017;12(2):213.
[2] Van der Auwera GA, Carneiro MO, Hartl C, Poplin R, Del Angel G, Levy-Moonshine A, et al. From FASTQ data to high-confidence variant calls: the genome analysis toolkit best practices pipeline. Curr Protoc Bioinform 2013;43(1). 11.10.1-11.10.33.
[3] Kessner D, Chambers M, Burke R, Agus D, Mallick P. ProteoWizard: open source software for rapid proteomics tools development. Bioinformatics 2008;24(21):2534–6.
[4] Chen S, Zhou Y, Chen Y, Gu J. Fastp: an ultra-fast all-in-one FASTQ preprocessor. Bioinformatics 2018;34(17):i884–90.
[5] Liu Y, Zhang LY, Li J. Fast detection of maximal exact matches via fixed sampling of query K-mers and Bloom filtering of index K-mers. Bioinformatics 2019;35(22):4560–7.
[6] Grabowski S, Bieniecki W. CopMEM: finding maximal exact matches via sampling both genomes. Bioinformatics 2019;35(4):677–8.
[7] Malysa G, Hernaez M, Ochoa I, Rao M, Ganesan K, Weissman T. QVZ: lossy compression of quality values. Bioinformatics 2015;31(19):3122–9.
[8] Liu Y, Yu Z, Dinger ME, Li J. Index suffix–prefix overlaps by (w, k)-minimizer to generate long contigs for reads compression. Bioinformatics 2019;35(12):2066–74.
[9] Chandak S, Tatwawadi K, Ochoa I, Hernaez M, Weissman T. SPRING: a next-generation compressor for FASTQ data. Bioinformatics 2019;35(15):2674–6.
[10] Pinho AJ, Garcia SP, Pratas D, Ferreira PJ. DNA sequences at a glance. PLoS One 2013;8(11):e79922.
[11] Pinho AJ, Pratas D, Ferreira PJ, Garcia SP. Symbolic to numerical conversion of DNA sequences using finite-context models. In: 2011 19th European signal processing conference. IEEE; 2011, p. 2024–8.
[12] Pratas D, Hosseini M, Pinho AJ. GeCo2: an optimized tool for lossless compression and analysis of DNA sequences. In: International conference on practical applications of computational biology & bioinformatics. Springer; 2019, p. 137–45.
[13] Hosseini M, Pratas D, Pinho AJ. AC: a compression tool for amino acid sequences. Interdiscip Sci: Comput Life Sci 2019;11(1):68–76.
[14] Carvalho JM, Brás S, Pratas D, Ferreira J, Soares SC, Pinho AJ. Extended-alphabet finite-context models. Pattern Recognit Lett 2018;112:49–55.
[15] Hosseini M, Pratas D, Pinho A. A survey on data compression methods for biological sequences. Information 2016;7(4):56.
[16] Agüero-Chapin G, González-Díaz H, Molina R, Varona-Santos J, Uriarte E, González-Díaz Y. Novel 2D maps and coupling numbers for protein sequences. The first QSAR study of polygalacturonases; isolation and prediction of a novel sequence from Psidium guajava L.. FEBS Lett 2006;580(3):723–30.
[17] Pratas D, Silva RM, Pinho AJ, Ferreira PJ. An alignment-free method to find and visualise rearrangements between pairs of DNA sequences. Sci Rep 2015;5:10203.
[18] Hosseini M, Pratas D, Morgenstern B, Pinho AJ. Smash++: an alignment-free and memory-efficient tool to find genomic rearrangements. GigaScience 9(5).
[19] Zielezinski A, Girgis HZ, Bernard G, Leimeister C-A, Tang K, Dencker T, et al. Benchmarking of alignment-free sequence comparison methods. 2019, 611137, BioRxiv.
[20] Forslund SK, Kaduk M, Sonnhammer EL. Evolution of protein domain architectures. In: Evolutionary genomics. Springer; 2019, p. 469–504.
[21] Rost B. Twilight zone of protein sequence alignments. Protein Eng 1999;12(2):85–94.
[22] Pratas D, Pinho AJ, Silva RM, Rodrigues JM, Hosseini M, Caetano T, et al. FALCON: a method to infer metagenomic composition of ancient DNA. 2018, 267179, BioRxiv.
[23] Pratas D, Pinho AJ. Metagenomic composition analysis of sedimentary ancient DNA from the isle of wight. In: 2018 26th european signal processing conference (EUSIPCO). IEEE; 2018, p. 1177–81.
[24] Huang W, Li L, Myers JR, Marth GT. ART: a next-generation sequencing read simulator. Bioinformatics 2011;28(4):593–4.
[25] Droop AP. Fqtools: an efficient software suite for modern FASTQ file manipulation. Bioinformatics 2016;32(12):1883–4.
[26] Afgan E, Baker D, Batut B, Van Den Beek M, Bouvier D, Čech M, et al. The galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic Acids Res 2018;46(W1):W537–44.
[27] Gordon A, Hannon G, et al. Fastx-toolkit, FASTQ/A short-reads preprocessing tools http://hannonlab.cshl.edu/fastx_toolkit Accessed: 2020-06-17.
[28] Shen W, Le S, Li Y, Hu F. SeqKit: a cross-platform and ultrafast toolkit for FASTA/Q file manipulation. PLoS One 2016;11(10):e0163962.
[29] DePristo MA, Banks E, Poplin R, Garimella KV, Maguire JR, Hartl C, et al. A framework for variation discovery and genotyping using next-generation DNA sequencing data. Nature Genet 2011;43(5):491.

[30] Goecks J, Nekrutenko A, Taylor J. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol 2010;11(8):R86.
[31] Blankenberg D, Gordon A, Von Kuster G, Coraor N, Taylor J, Nekrutenko A, et al. Manipulation of FASTQ data with galaxy. Bioinformatics 2010;26(14):1783–5.
[32] Liu Y, Peng H, Wong L, Li J. High-speed and high-ratio referential genome compression. Bioinformatics 2017;33(21):3364–72.
[33] Ochoa I, Hernaez M, Weissman T. iDoComp: a compression scheme for assembled genomes. Bioinformatics 2014;31(5):626–33.
[34] Deorowicz S, Danek A, Niemiec M. GDC 2: Compression of large collections of genomes. Sci Rep 2015;5:11565.
[35] Hernaez M, Pavlichin D, Weissman T, Ochoa I. Genomic data compression. Annu Rev Biomed Data Sci 2.
[36] Pratas D, Hosseini M, Pinho AJ. Compression of amino acid sequences. In: International conference on practical applications of computational biology & bioinformatics. Springer; 2018, p. 105–13.
[37] Pratas D, Pinho AJ, Rodrigues JM. XS: a FASTQ read simulator. BMC Res Notes 2014;7(1):40.
[38] Escalona M, Rocha S, Posada D. A comparison of tools for the simulation of genomic next-generation sequencing data. Nature Rev Genet 2016;17(8):459.